Data-Driven Identification of Main Behavioural Classes and Characteristics of Resident Space Objects in Low Earth Orbit through Unsupervised Learning

Marta Guimarães
Neuraspace; FCT-UNL
marta.guimaraes@neuraspace.com

Cláudia Soares
FCT-UNL
claudia.soares@fct.unl.pt

Chiara Manfletti
Neuraspace
chiara.manfletti@neuraspace.com

ABSTRACT

In recent years, the number of objects in low Earth orbit (LEO) has witnessed a significant increase, which poses a challenge to decision-making and overall space situational awareness. To make critical decisions such as collision avoidance manoeuvres, satellite owners and operators must be aware of objects in close proximity to their assets. To achieve this, it is crucial to understand the characteristics of such resident space objects (RSOs). In this paper, we analyse a large dataset of catalogued objects and publicly available two-line element data to identify the main characteristics of RSOs in LEO. We use unsupervised learning techniques to find patterns in the data and group the RSOs by their historical orbital information and patterns of life. Specifically, we use a novel deep learning-based clustering algorithm to identify essential clusters leveraging the dynamic information of the RSOs. We further explore the obtained clusters by using dimensionality reduction techniques, providing deeper insights into the dynamics of objects in orbit. We discuss the limitations of current satellite datasets in providing comprehensive information, such as the level of automation of RSOs or classification criteria for new satellite shapes. Our work contributes to a better understanding of RSOs in LEO by uncovering patterns and identifying fundamental clusters of RSOs based on their characteristics, therefore improving space situational awareness and decision-making for satellite owners and operators.

1. INTRODUCTION

The exponential increase in the number of resident space objects (RSOs) [1] has become a major concern [2–4]. With the increasing orbital population, space traffic management (STM) and space situational awareness (SSA) have become more complex, creating challenges for satellite operators, especially in ensuring collision avoidance.

The main challenge lies in the current methods to track, determine and propagate the orbits of RSOs, which are not as effective as desired [5]. Consequently, identifying the operational patterns and behaviours, or patterns of life (PoL), of RSOs also becomes a significant challenge. PoL refer to the typical behaviours and movements of RSOs over time, e.g., station-keeping and collision avoidance manoeuvres. Understanding these patterns is crucial for distinguishing between nominal (expected) and anomalous (unexpected or potentially hazardous) behaviours. This knowledge enables satellite operators to make informed decisions about manoeuvring satellites to avoid collisions. However, this task is further exacerbated by the increasing variety and number of RSOs, highlighting the critical need for more robust methodologies. As the population and complexity of RSOs grow, current tracking and characterisation methods fall short, limiting our ability to maintain space safety and operational efficiency.

This study addresses these pressing needs by presenting a new approach to extracting meaningful information from RSOs' temporal patterns in low-Earth orbit (LEO).

1.1 Related Work

The problem of identifying PoL and behavioural modes has been explored by several researchers. They have primarily used machine learning techniques to address this challenge, often leveraging either unsupervised [6–9] or supervised learning [10–19], approaches. Statistical techniques have also been explored [20, 21], offering another dimension to the analysis. A challenge competition has also been organised [22], with the goal of exploring the use of AI algorithms to uncover PoLs in geostationary orbit (GEO), and to provide a labelled dataset of satellite's manoeuvres to the community.

Indeed, one of the main challenges around the problem of detecting RSOs' PoLs, is the lack of information about the satellite manoeuvres, which are of major importance to classify the operational behaviours. Unfortunately, it is not possible to find a comprehensive catalogue of manoeuvres due to data-sharing agreements, privacy, and security reasons. In LEO, this issue is even more pronounced due to the increasing number of objects in orbit and the corresponding frequency of manoeuvres performed on a daily basis.

Even if such manoeuvre data were to be partially made publicly accessible, achieving good performance in operational behaviour classification would require a considerable amount of high-quality labelled data. This presents another significant challenge, as obtaining quality labels in sufficient quantities is often not feasible.

For this reason, using an unsupervised learning approach is a logical and practical choice. By relying on this method, we do not depend on labelled data and instead, aim to uncover the patterns directly from the data.

As already mentioned, several studies have leveraged unsupervised learning models to identify patterns in satellite data. Mital et al. [6] explored different clustering techniques to group the two-line element (TLE) data and find patterns, particularly focusing on GEO, using only latitude and longitude with respect to time as features, concerning a single day of data. Bai et al. [7] also attempted to cluster TLE data according to the magnitude of orbital manoeuvres (i.e., large, medium, and small-scale) by using a technique that combines k-means clustering with contour mapping. Roberts et al. [8, 9] transformed GEO satellite position time-histories into short segments and sorted them into groups of similar features using time-series k-means clustering.

1.2 Contributions

While previous works have explored the use of different unsupervised learning models to understand RSOs' PoL, none of them focus on exploring historical orbital information from LEO. Thus, we propose a new approach using unsupervised learning techniques to find such patterns in the data. Specifically, we use a novel deep learning-based clustering algorithm to identify essential clusters leveraging the historical dynamic information of the RSOs. We further explore the obtained clusters by using dimensionality reduction techniques, providing deeper insights into the dynamics of objects in orbit. Additionally, we take advantage of the inherent characteristics of the RSOs to gain a better understanding of the resulting clusters. We discuss the limitations of current satellite datasets in providing comprehensive information, such as the level of automation of RSOs or classification criteria for new satellite shapes. Our work focuses on the LEO, due to the increasing population of RSOs in this orbit.

2. TWO-LINE ELEMENT DATA

We use publicly available TLE data from the U.S. Space Command's space object catalogue [1]. Such a catalogue includes RSOs' historical orbital parameters, which can be used to describe long-term changes in their orbital trajectories throughout their lifetimes. The key parameters of the TLEs include:

- Inclination (degrees): The angle between the satellite's orbital plane and the Earth's equator, determining the satellite's orbital tilt and coverage pattern.
- Right Ascension of the Ascending Node (RAAN) (degrees): The angle from a reference direction (usually the vernal equinox) to the point where the satellite crosses the equatorial plane heading north. RAAN specifies the orbit's orientation in the equatorial plane.
- Eccentricity (decimal point): Describes the shape of the orbit, with values ranging from 0 (circular) to close to 1 (highly elliptical). Eccentricity affects the satellite's altitude variation during orbit.

[1] space-track.org

- Argument of Perigee (degrees): The angle within the orbital plane from the ascending node to the orbit's closest approach to Earth (perigee), indicating where the satellite's lowest altitude occurs.
- Mean Anomaly (degrees): A parameter that helps determine the satellite's position along its orbit at a given time. It represents the fraction of an orbit's period that has elapsed since the satellite passed the perigee.
- Mean Motion (revolutions per day): The average number of orbits the satellite completes around Earth in a day, directly related to the orbital period and altitude.

Given that features expressed in degrees (inclination, RAAN, argument of perigee, and mean anomaly) exhibit discontinuities at 0°/180° or 0°/360°, an angle wrap-around point was used to ensure continuous representation across these boundaries.

Considering the scope of our problem, only samples from RSOs in LEO were used. Besides, due to the computational constraints of this work, we have chosen to limit the range of data to six months, corresponding to the first months of 2023.

3. METHOD

In this study, we address the challenge of clustering high-dimensional time series data of RSOs in LEO, where traditional methods like k-means and dynamic time warping [23] failed to produce meaningful clusters. To overcome this limitation, we developed a deep learning-based approach to learn more robust and informative representations of the time series data before applying clustering. More concretely, we used masked autoencoders, a state-of-the-art self-supervised approach [24, 25].

Using a masked autoencoder to learn representations of a time series for downstream tasks (in our case, clustering) is analogous to the principles behind masked language and image modelling. In the same way that masked language models like Bidirectional Encoder Representations from Transformers (BERT) [24] predict missing words to capture deep contextual representations of text, and masked image models [25] reconstruct missing parts of images to learn intricate visual features, a masked autoencoder for time series data trains a neural network by masking portions of the input sequence and reconstructing them.

However, there is a difference in information density between language and vision, requiring different modelling strategies. Language signals are highly semantic and information-dense, meaning that even a few missing words in a sentence may lead to sophisticated language understanding. This is because each word carries significant meaning and context. In contrast, images are natural signals with heavy spatial redundancy, where missing patches can often be inferred from neighbouring patches without requiring a deep understanding of the overall scene. To encourage models to learn useful features from images, a common strategy is to mask a large portion of random patches, thereby reducing redundancy and creating a challenging self-supervised task that requires a holistic understanding beyond low-level image statistics.

Time series data, like images, often exhibit redundancy, especially in segments with consistent patterns. By exploiting the similarity between the redundancy found in time series and images we introduce substantial masking to large portions of the data thereby creating a more challenging task that forces the model to learn comprehensive and meaningful representations. This approach encourages the model to understand the broader temporal dependencies and structures within the data, rather than relying on local, redundant information. Classical methods for time series analysis, which typically use engineered features and simpler models, may not capture these complex dependencies as effectively. Thus, we opted to use a vision-inspired masking strategy rather than applying classical time series representation methods. We propose a novel algorithm inspired by the stable state-of-the-art model for representation learning in images, the Masked Autoencoder (MAE) [25] (Fig. 1).

3.1 Masked Autoencoder

The model proposed by He et al. [25] is a simple autoencoding approach that reconstructs the original signal based on a partial observation. Like all autoencoders, it has an encoder that maps the observed signal to a latent representation, and a decoder that rebuilds the original signal from the latent representation. Unlike previously proposed autoencoders, the authors used an asymmetric design that allows the encoder to work only on the partial, observed signal (without mask tokens) and a decoder that reconstructs the full signal from the latent representation and mask tokens.

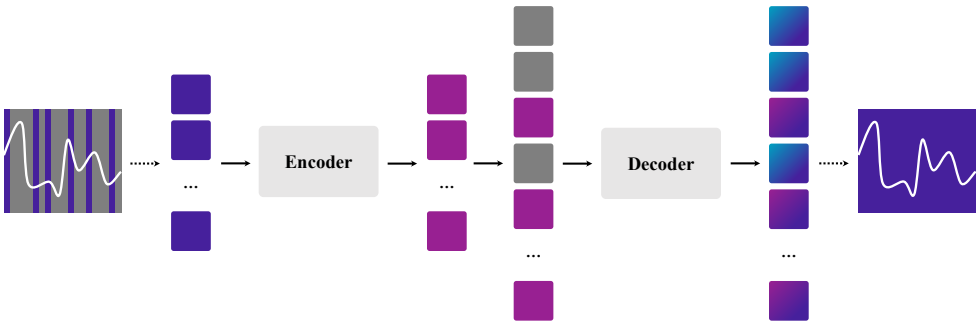

Fig. 1: **Time masked autoencoder.** A large random subset of patches (e.g., 75%) is masked. The encoder is applied to the subset of unmasked tokens. A decoder processes the full set of encoded and masked tokens to reconstruct the original time series in steps. (i) purple: original data; (ii) grey: masked attributes; (iii) pink: embedding of unmasked attributes; (iv) gradients: reconstruction of original data. Figure adapted from [25].

Masking. Similar to the MAE, we follow the Vision Transformer (ViT) [26] approach, i.e., we divide each time series into regular and non-overlapping patches. Then, we sample a random subset of patches and mask them. Given the information density property already mentioned, such sampling is performed with a high masking ratio. This highly sparse input creates the opportunity to have a larger encoder.

Encoder. The encoder only processes the unmasked patches. It embeds patches using a linear projection, adds positional embeddings, and processes the set with Transformer blocks [27], similar to a standard ViT.

Decoder. The decoder takes in a complete set of tokens, i.e., encoded unmasked patches and mask tokens. Each mask token is a learned vector that shows where a missing patch needs to be predicted. Positional embeddings are also added to the masked tokens, and a series of Transformer blocks are used to process the data. Note that the decoder is only used during training (Fig. 2).

Reconstruction Task. The loss is computed between the reconstructed and the original time series in the space of the time steps.

`[CLS]` **Token.** Similar to the BERT and ViT setups, an extra learnable token is appended to the input sequence. During the processing of the input sequence, the `[CLS]` token aggregates information from all other tokens in the sequence, essentially acting as a summary representation of the entire input.

Clustering. After training, the decoder is discarded and the encoder is applied to unmasked time series for the clustering task (Fig. 2). We obtain the encoded representation of the `[CLS]` token for each time series and perform clustering on the `[CLS]` embedding space.

4. EXPERIMENTS

4.1 Ablation Studies

As discussed by He et al. [25] in the original MAE architecture, the encoder operates on a smaller subset of the full set, e.g., 25%, as masked patches are not considered. In principle, this allows us to consider larger encoders. On the other hand, the input to the decoder already considers the full data set, as masked patches are used. Thus, smaller decoders should be used. With this asymmetrical design, the full set of tokens is only processed by the lightweight decoder, which reduces training time.

To validate this asymmetrical design, we have performed some ablation experiments. Table 1 shows the results obtained when considering the best validation loss from several trials to ensure a fair comparison. The results are presented as (mean ± std) loss values obtained from such trials.

The first analysis (Table 1a) aimed to understand the impact of considering a higher number of encoder layers when

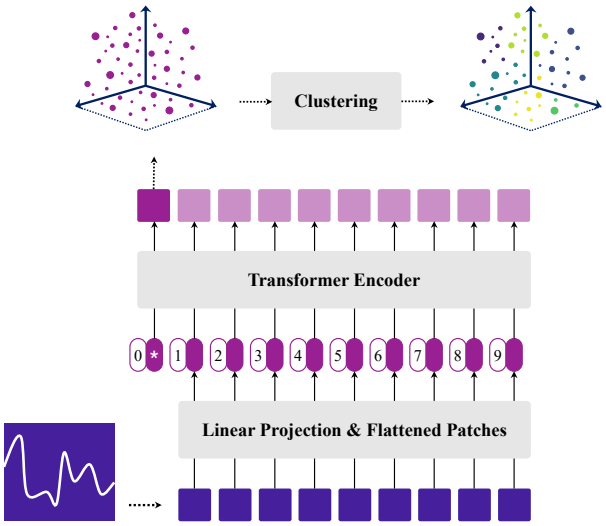

Fig. 2: **Inference setup.** After training, the decoder is discarded and the encoder is applied to an unmasked time series for the clustering task. "*" stands for the extra learnable `[CLS]` embedding. Figure adapted from [26].

compared to the default value of 2 decoder layers. Surprisingly, all the setups achieved a loss value within the same order of magnitude. The mean values contradicted the expected behaviour, suggesting that larger encoders do not necessarily lead to better performance. To further verify these results, we experimented with changing the number of decoder layers, while keeping the asymmetrical design. Table 1b, shows that, once again, the loss values achieved were within the same order of magnitude.

We have also explored the impact of considering different embedding dimensions (Table 1c), masking ratios (Table 1d), and patch size (Table 1e). As shown in the referred tables, the obtained loss values remain within the same order of magnitude as previously.

It is important to highlight that our evaluation focused on the loss function rather than alternative metrics due to the nature of our dataset, which includes features with widely varying ranges of values. Evaluating reconstruction performance could introduce bias, favouring certain features over others. Instead, by relying on the loss, we ensure a more balanced assessment of the model's ability to learn meaningful representations, which is, in fact, our main goal.

As seen in Table 1, the range of model complexities varied significantly. Despite this variation, the loss values remained within the same order of magnitude across all hyperparameter settings. This consistent performance indicates that the model is relatively insensitive to the hyperparameters within the tested ranges. This robustness suggests that the model's architecture and training process are inherently stable, providing reliable performance without requiring precise tuning of hyperparameters.

Given the limited sensitivity to hyperparameters, the results also imply that our current model configuration is potentially close to optimal for the given dataset and task. The model appears to have sufficient capacity to learn the underlying data distribution effectively, as evidenced by the stable loss values across different trials. Expanding the hyperparameter search space or considering alternative model architectures may yield further insights, but the current findings suggest a solid baseline performance.

5. POST-TRAINING CLUSTERING

Since the goal of this work is to cluster the RSOs' historical patterns, at this stage, the entire dataset used to train and evaluate the model is considered. Thus, after training the model, the full dataset is used in the setup described in Fig. 2. This means each sample will have a learned embedding representation for each patch and the `[CLS]` token.

Table 1: **Architecture ablation studies.** The results were obtained from different trials to ensure a fair comparison. The results are presented as (mean ± std). If not specified, the default is: 8 encoder layers, 2 decoder layers, embedding dimension of 16, 75% masking ratio, patch size of 8, and 4 heads for both the encoder and the decoder.

(a) Encoder layers.

Encoder	Loss
2	(1.98 ± 0.05)e-1
4	(2.17 ± 0.24)e-1
8	(2.39 ± 0.08)e-1
16	(2.41 ± 0.06)e-1

(b) Large encoder and light decoder.

Encoder	Decoder	Loss
8	2	(2.39 ± 0.08)e-1
8	4	(2.39 ± 0.07)e-1
16	6	(3.06 ± 1.02)e-1
16	8	(3.74 ± 1.03)e-1

(c) Embedding dimension.

d_{model}	Loss
16	(2.39 ± 0.08)e-1
32	(1.78 ± 0.19)e-1
64	(1.53 ± 0.09)e-1

(d) Masking ratio.

Masking	Loss
70%	(2.39 ± 0.03)e-1
75%	(2.39 ± 0.08)e-1
80%	(2.30 ± 0.02)e-1
85%	(2.41 ± 0.08)e-1

(e) Patch size.

Patch	Loss
4	(2.31 ± 0.08)e-1
8	(2.39 ± 0.08)e-1
16	(2.36 ± 0.04)e-1

The next step is to perform clustering on the embedding space of the `[CLS]` tokens. To eliminate the need for predefining the number of clusters or relying on methods like the within-cluster sum of squares, we opted for Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [28]. Unlike traditional clustering techniques, DBSCAN identifies clusters based on the density of data points. It detects core samples within dense regions and expands clusters outward from these cores, capturing clusters of varying shapes and handling outliers.

The DBSCAN algorithm is sensitive to its core parameters, particularly the epsilon (ε) parameter, which defines the maximum distance between two points for them to be considered as part of the same cluster, and `min_samples`, which, as the name suggests, defines the minimum number of points required to form a cluster. These parameters play a crucial role in determining the granularity of the clusters.

Fig. 3 presents the resulting clusters obtained when exploring the impact of such parameters. The clusters were obtained when using the best-performing model from validation.

However, since the `[CLS]` embedding space where the clustering was performed is high-dimensional (Table 1c), a dimensionality reduction technique is needed to visualise such data. In other words, to visualise the resulting clusters, we need to project the data from d_{model}-dimensional space to a lower-dimensional space, i.e. two-dimensional (2D). Thus, we have used the Uniform Manifold Approximation and Projection (UMAP) [29] algorithm due to its advantages when compared to other methods [29].

Fig. 3 shows the 2D projection of the `[CLS]` embedding space via UMAP. It can be concluded that:

- When ε is set to a higher value, DBSCAN tends to group more data points into fewer, larger clusters. This broad clustering can obscure smaller, more specific groupings, as the algorithm is more likely to include objects with general similarities in their orbital parameters.
- As ε is decreased, the algorithm becomes more sensitive to subtle differences in the data, resulting in a higher number of smaller, more specific clusters. Notably, when ε is sufficiently reduced, DBSCAN starts to identify clusters that correspond to satellite constellations—groups of satellites that share nearly identical orbital parameters. These clusters emerge because a smaller ε captures the fine distinctions between closely spaced orbits typical of constellation satellites.
- A higher `min_samples` value means that more data points are needed to form a cluster, which can lead to fewer but more robust clusters. This setting reduces the likelihood of forming small clusters, instead emphasising larger, denser groups that are more consistent across the dataset.
- Decreasing the `min_samples` value allows DBSCAN to recognise smaller, potentially more numerous clusters.

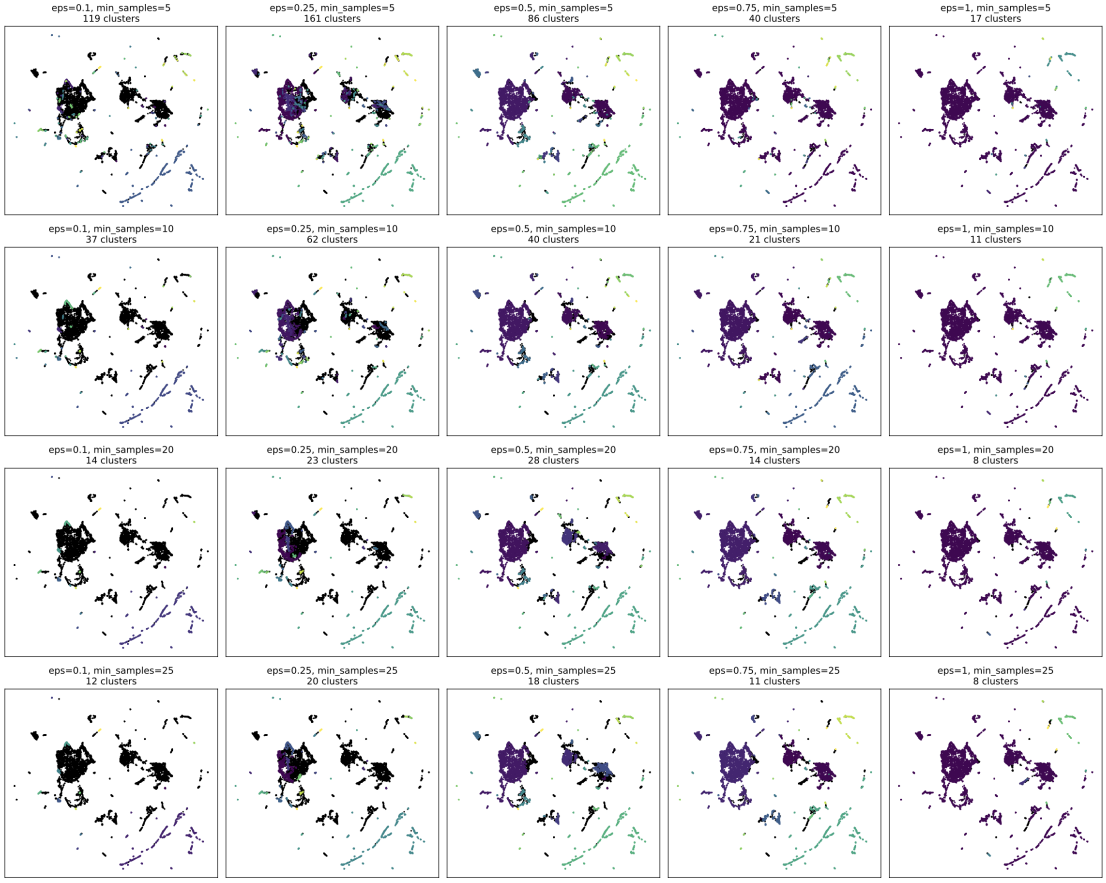

Fig. 3: **2D projection of the `[CLS]` embedding space via UMAP.** The colours represent the clusters obtained when applying the DBSCAN with different parameters. Black samples denote the noise values, i.e., samples that were not associated with any cluster.

> This can be particularly useful when trying to detect smaller groups, such as specific satellite constellations or less common orbital configurations, where fewer points share close proximity.

To further explore the resulting clusters, and given the trade-offs already discussed, we fix the ε and `min_samples` parameters as 0.5 and 5, respectively (Fig. 4). As already seen in Fig. 3, such parameters result in 86 clusters.

Upon initial examination of Fig. 4, it is evident that two clusters notably stand out from the others. These clusters, specifically Cluster 5 (shown in purple) and Cluster 59 (in green), correspond to the clusters with the highest number of samples, representing approximately 32% and 14% of the data, respectively. Due to the large number of samples in these clusters, when analysing the orbital elements used to group the data, no outstanding patterns can be extracted. Such results suggest that these clusters share common but non-specific behaviours or that the clustering method reached its limit in differentiating subtle variations in the data. However, to get further understanding of such clusters, we used information from the Database and Information System Characterising Objects in Space (DISCOS)[2] [30]. This dataset provides information regarding the key characteristics of objects, including their size, mass and object type.

When analysing the RSOs that compose Cluster 5, it can be concluded that according to DISCOS, approximately 73% of the data are debris, 3% are rocket-bodies, 0.4% are unknown, and the remaining are payloads. Orbital debris, while generally passive and lacking propulsion, can exhibit unpredictable orbital characteristics due to factors like collisions, fragmentation, and irregular shapes. These variations can lead to complex orbital dynamics, influenced

[2] discosweb.esoc.esa.int

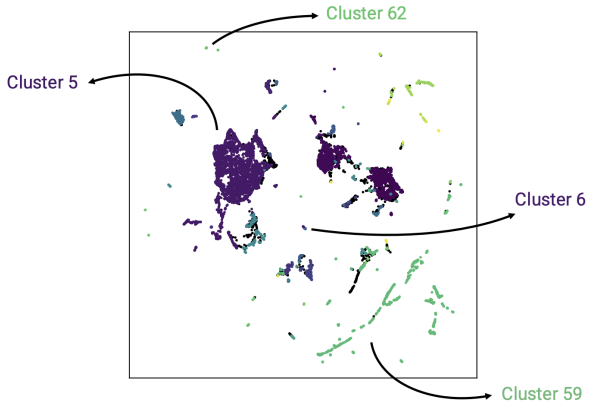

Fig. 4: **Resulting clusters for a given set of DBSCAN parameters.** ϵ and `min_samples` equal to 0.5 and 5, respectively.

by external forces such as atmospheric drag and gravitational perturbations. This unpredictability might contribute to the challenges in identifying distinct patterns within this cluster, as the debris objects, despite sharing a common classification, may display diverse and evolving orbital behaviours.

As for Cluster 59, approximately 99% of the objects are identified as payloads, with nearly 99% of these belonging to the Starlink constellation [3]. Remarkably, this cluster alone accounts for approximately 67% of all Starlink satellites present in our dataset. The consistency in the formation of this cluster, even when varying the parameters in the DBSCAN algorithm (Fig. 3), underscores the robustness of this grouping. Further analysis revealed that smaller clusters within the dataset are also composed entirely of Starlink satellites, indicating that the algorithm is sensitive enough to distinguish between subtle differences within the constellation. In this case, the difference lies in the fact that the satellites in these smaller clusters were launched after those in the main cluster. These findings suggest that the model is highly effective at recognising and clustering based on distinct orbital patterns and operational behaviours, particularly within large satellite constellations like Starlink.

In addition to the larger clusters dominated by constellations like Starlink, the analysis also identified smaller, more specialised clusters. One such cluster, Cluster 6 in Fig. 4, consists of debris from the SNAP-10A mission (code-named "Snapshot"), as well as the SECOR 4 satellite. Interestingly, SECOR 4 was launched as a secondary payload alongside SNAP-10A. This cluster's composition highlights the algorithm's capability to group objects based on shared launch histories and resulting orbital dynamics. The presence of both SNAP-10A debris and SECOR 4 within the same cluster suggests that these objects, despite their different purposes, share similar orbital characteristics due to their common origin. The debris from SNAP-10A likely follows a dispersal pattern influenced by the original orbital parameters of the launch vehicle, which also carried SECOR 4. The identification of this specific cluster demonstrates the model's sensitivity to the nuances of orbital behaviour that arise from mission-specific events, such as the fragmentation of a satellite. By clustering these objects together, the algorithm effectively captures the relationship between the primary and secondary payloads and their subsequent evolution in space. This ability to discern such fine details underscores the model's strength in identifying and analysing less common orbital patterns that may not be immediately apparent in broader clustering efforts.

Another example of a smaller cluster identified by the model consists of a subset of satellites from the OneWeb constellation [4] (Cluster 62 in Fig. 4). Fig. 5 illustrates the evolution of the orbital elements over time. The subtle changes in the orbital parameters, especially the mean motion, suggest that some satellites in this cluster might have undergone minor orbital adjustments or station-keeping manoeuvres. These are typical for constellations like OneWeb, where satellites regularly adjust their orbits to maintain precise spacing and coverage patterns. The clustering ability of the model to detect these variations further demonstrates its effectiveness in not only distinguishing between different groups of satellites but also identifying potential operational activities, such as manoeuvres, within those groups.

[3] starlink.com
[4] oneweb.net

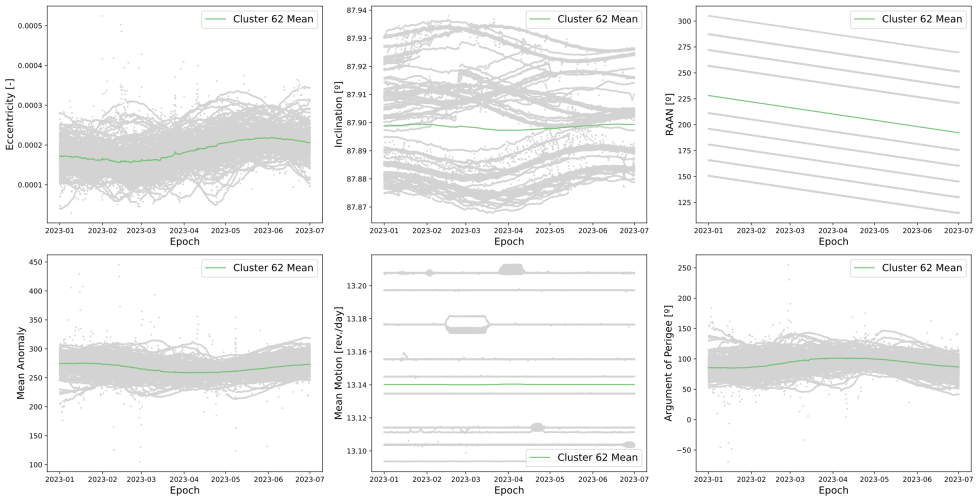

Fig. 5: **Cluster 62.** Each grey line represents the evolution of the orbital parameters for a given RSO.

However, it is important to recall that the formation of such specific clusters is directly influenced by the choice of parameters in the clustering algorithm. Adjusting these parameters, such as the epsilon ε and `min_samples` values in DBSCAN, can significantly change the model's ability to capture certain patterns in the data. When such parameters are set to produce a higher number of clusters, the algorithm is more likely to detect finer, more detailed patterns within the data. This typically results in the identification of individual satellite constellations or even subsets of constellations, as the model becomes sensitive to subtle differences in orbital elements and operational behaviours. When the parameters are adjusted to produce fewer clusters, the model tends to capture broader, more generalised operational patterns. In this scenario, specific operational details or nuances may be combined, emphasising only the overarching trends and not the intricate details.

6. LIMITATIONS OF CURRENT SATELLITE DATASETS

The proposed clustering model has proven to be a highly valuable tool in analysing and grouping RSOs based solely on TLEs. Despite the inherent limitations of using only TLE data, the model effectively identified distinct patterns and behaviours, highlighting its robustness and potential in SSA. The ability of the model to differentiate between various satellite constellations, detect potential manoeuvres, and classify clusters based on orbital dynamics speaks to the power of this approach.

However, while TLEs offer critical information regarding the orbital elements of RSOs, they represent just one dimension of a much broader and more complex picture. To gain a more comprehensive understanding of satellite operations and behaviours, the next logical step is to incorporate additional data sources. These could include satellite telemetry, mission-specific details, ground-based observations, and other relevant datasets that provide insights into factors beyond the scope of TLEs.

Despite the clear benefits of integrating additional data sources, there are significant limitations related to data availability. Some crucial information, which would greatly improve our understanding of RSOs, is not publicly accessible or is incomplete. For example, the degree of automation in satellite operations significantly impacts behaviour, especially regarding manoeuvring and station-keeping. However, comprehensive data on the automation levels is often not accessible to the public, hindering full comprehension of the decision-making processes involved in observed orbital changes. It is also important to consider the physical characteristics and shapes of satellites. As satellite designs become more varied, a challenge arises due to the lack of comprehensive classification criteria for new satellite shapes and structures. This information is crucial for accurately predicting orbital behaviour, particularly in LEO where atmospheric drag can have a significant impact on satellites with different shapes.

Overcoming these limitations will require collaboration across the space industry, including data-sharing agreements

and the development of new observational techniques. By broadening the data inputs, we can refine our clustering models even further, ultimately contributing to safer and more sustainable space operations.

7. CONCLUSION

We introduced a novel approach for clustering time series orbital data, i.e., TLEs, using a vision-inspired masking representation learning model. This innovative technique allowed us to capture and represent the complex temporal patterns of such high-dimensional data. Subsequently, we applied a clustering model to group these time series into distinct clusters, enabling a deeper understanding of the underlying behaviours and operational characteristics of various RSOs. Our analysis also explored the impact of considering different clustering parameters. More concretely, we demonstrate and discuss how adjustments in parameters such as the ε and `min_samples` values in DBSCAN can lead to the identification of different levels of patterns within the data. It is important to note that the choice of such parameters is dependent on the desired application, as changing these parameters can lead to detecting either broad operational trends or more refined, constellation-specific behaviours, highlighting the flexibility and adaptability of our approach.

The results of our clustering model revealed significant value for SSA and STM. The ability to automatically classify and group RSOs based on their orbital patterns provides a powerful tool for monitoring satellite constellations and understanding the operational behaviours of both active satellites and debris. For instance, by categorising satellites into groups based on shared characteristics, orbit determination and propagation algorithms can be tailored to account for the common properties within each group. More concretely, satellites within the same constellation or debris cluster may exhibit similar drag profiles or orbital decay rates, allowing for more precise and efficient predictions of their future positions.

This approach offers a scalable solution for managing the increasingly crowded space environment and contributes to the development of more effective strategies for ensuring the long-term sustainability of space activities.

Future work should focus on integrating additional data sources to further refine the clustering results and address some of the limitations discussed, ultimately improving our ability to maintain a safe and secure space environment.

8. ACKNOWLEDGEMENTS

This research was carried out under Project "Artificial Intelligence Fights Space Debris" Nº C626449889-0046305 co-funded by Recovery and Resilience Plan and NextGeneration EU Funds (www.recuperarportugal.gov.pt), and by NOVA LINCS (UIDB/04516/2020) with the financial support of FCT.IP.

9. REFERENCES

[1] ESA Space Debris Office. ESA'S Annual Space Environment Report. 2024.

[2] Donald J. Kessler and Burton G. Cour-Palais. Collision Frequency of Artificial Satellites: The Creation of a Debris Belt. *Journal of Geophysical Research: Space Physics*, 83:2637–2646, 1978. doi: 10.1029/JA083iA06p02637.

[3] B. Bastida Virgili, J. C. Dolado, H. G. Lewis, J. Radtke, H. Krag, B. Revelin, C. Cazaux, C. Colombo, R. Crowther, and M. Metz. Risk to Space Sustainability from Large Constellations of Satellites. *Acta Astronautica*, 126:154–162, 2016. doi: 10.1016/j.actaastro.2016.03.034.

[4] Benjamin Bastida Virgili, Tim Flohrer, Holger Krag, Klaus Merz, and Stijn Lemmens. CREAM: ESA's Proposal for Collision Risk Estimation and Automated Mitigation. *1st International Orbital Debris Conference*, 2019.

[5] Francisco Caldas and Cláudia Soares. Machine learning in orbit estimation: A survey. *Acta Astronautica*, 220: 97–107, 2024. ISSN 0094-5765. doi: https://doi.org/10.1016/j.actaastro.2024.03.072. URL `https://www.sciencedirect.com/science/article/pii/S0094576524001917`.

[6] Rohit Mital, Kim Cates, Joe Coughlin, and Geetha Ganji. A Machine Learning Approach to Modeling Satellite Behavior. In *2019 IEEE International Conference on Space Mission Challenges for Information Technology (SMC-IT)*, pages 62–69, 2019. doi: 10.1109/SMC-IT.2019.00013.

[7] Xue Bai, Chuan Liao, Xiao Pan, and Ming Xu. Mining Two-Line Element Data to Detect Orbital Maneuver for Satellite. *IEEE Access*, 7:129537–129550, 2019. doi: 10.1109/ACCESS.2019.2940248.

[8] Thomas Roberts, Victor Rodriguez-Fernandez, Peng Mun Siew, Haley Solera, and Richard Linares. End-to-end behavioral mode clustering for geosynchronous satellites. In *24th Advanced Maui Optical and Space Surveillance Technologies*, 09 2023.

[9] Thomas Roberts, Haley Solera, and Richard Linares. Geosynchronous satellite behavior classification via unsupervised machine learning. In *9th IAA Space Traffic Management Conference*, 03 2023.

[10] Charlotte Shabarekh, Jordan Kent-Bryant, Gene Keselman, and Andonis Mitidis. A novel method for satellite maneuver prediction. In *Advanced Maui Optical and Space Surveillance Technologies Conference*, volume 24, 2016.

[11] Dan Shen, Jingyang Lu, Genshe Chen, Erik Blasch, Carolyn Sheaff, Mark Pugh, and Khanh Pham. Methods of Machine Learning for Space Object Pattern Classification). pages 565–572, 07 2019. doi: 10.1109/NAECON46414.2019.9058182.

[12] Nicholas Perovich, Zachary Folcik, and Rafael Jaimes. Satellite Maneuver Detection Using Machine Learning and Neural Network MethodsBehaviors. In *2022 IEEE Aerospace Conference (AERO)*, pages 01–19, 2022. doi: 10.1109/AERO53065.2022.9843412.

[13] Phil DiBona, James Foster, Anthony Falcone, and Michael Czajkowski. Machine learning for RSO maneuver classification and orbital pattern prediction. In *Advanced Maui Optical and Space Surveillance Technologies Conference*, page 54, 2019.

[14] Thomas Roberts and Richard Linares. Satellite Repositioning Maneuver Detection in Geosynchronous Orbit Using Two-line Element (TLE) Data. In *71st International Astronautical Congress*, 10 2020.

[15] Thomas Roberts. *Geosynchronous Satellite Maneuver Classification and Orbital Pattern Anomaly Detection via Supervised Machine Learning*. Master's thesis, Massachusetts Institute of Technology, 2021.

[16] Thomas G Roberts and Richard Linares. Geosynchronous satellite maneuver classification via supervised machine learning. In *Advanced Maui Optical and Space Surveillance Technologies Conference. Maui, Hawaii*, 2021.

[17] Riccardo Cipollone, Nugraha Setya Ardi, and Pierluigi Di Lizia. A supervised learning-based approach to maneuver detection through tle data mining. In *International Conference on Applied Intelligence and Informatics*, pages 419–434. Springer, 2022.

[18] Riccardo Cipollone, Italo Leonzio, Gaetano Calabrò, and Pierluigi Di Lizia. An lstm-based maneuver detection algorithm from satellites pattern of life. In *2023 IEEE 10th International Workshop on Metrology for AeroSpace (MetroAeroSpace)*, pages 78–83, 2023. doi: 10.1109/MetroAeroSpace57412.2023.10189993.

[19] Haley Solera, Thomas Roberts, and Richard Linares. Geosynchronous Satellite Pattern-of-Life Node Detection and Classification. In *9th IAA Space Traffic Management Conference*, 2023.

[20] John Bicknell, Paul Szymanski, and Werner Krebs. Space Object Pattern of Life Process Analysis, 10 2019.

[21] Shivshankar S and Debasish Ghose. Behaviour Modelling of Satellites for Space Situational Awareness using Time Series Analysis and k-Means Clustering. In *2021 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)*, pages 1–6, 2021. doi: 10.1109/CONECCT52877.2021.9622573.

[22] Peng Mun Siew, Haley Solera, Thomas Roberts, Daniel Jang, Victor Rodriguez-Fernandez, Jonathan How, and Richard Linares. Ai ssa challenge problem: Satellite pattern-of-life characterization dataset and benchmark suite. In *24th Advanced Maui Optical and Space Surveillance Technologies*, 09 2023.

[23] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49, 1978. doi: 10.1109/TASSP.1978.1163055.

[24] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *North American Chapter of the Association for Computational Linguistics*, 2019.

[25] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollar, and Ross Girshick. Masked Autoencoders Are Scalable Vision Learners. In *Proceedings - 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pages 15979–15988. IEEE Computer Society, 2022. doi: 10.1109/CVPR52688.2022.01553.

[26] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[27] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[28] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, page 226–231. AAAI Press, 1996.

[29] Leland McInnes, John Healy, Nathaniel Saul, and Lukas Großberger. UMAP: Uniform Manifold Approximation and Projection. *Journal of Open Source Software*, 3(29):861, 2018. doi: 10.21105/joss.00861. URL `https://doi.org/10.21105/joss.00861`.

[30] T. Flohrer, S. Lemmens, B. Bastida Virgili, H. Krag, H. Klinkrad, E. Parrilla, N. Sanchez, J. Oliveira, and F. Pina. DISCOS: Current Status and Future Developments. *6th European Conference on Space Debris*, 2013.

A. ABBREVIATIONS AND ACRONYMS

2D two-dimensional

BERT Bidirectional Encoder Representations from Transformers

DBSCAN Density-Based Spatial Clustering of Applications with Noise

DISCOS Database and Information System Characterising Objects in Space

GEO geostationary orbit

LEO low-Earth orbit

MAE Masked Autoencoder

PoL patterns of life

RAAN Right Ascension of the Ascending Node

RSO resident space object

SSA space situational awareness

STM space traffic management

TLE two-line element

UMAP Uniform Manifold Approximation and Projection

ViT Vision Transformer